Project Deliverable G:
Prototype II
GNG 1103
Prof Knox

Alex Milgram
Kalan Kok
Eric Wang
Charlie Huang
Peter Zhou

Table of Contents

1. Test Plan……..……..…3

2. Updated Target Specifications ……..…4

3. Critical Systems……..4-5

4. Prototype II……...6

5. Prototype III Test Plan…….……..6-10

6. References and Wrike ……...11

Test Plan

Transition of our ideas for project:

As per request of our client our client also wants to implement a way to gather speed in CPM from the conveyor; this can be done with a simple code with an arduino and a sensor.

Testing For Prototype 1:

Because we formatted our test plan for a different design we followed a different set of tests.

1. Ensure that the Code can Read the Data Properly

We tested this by imputing some value and seeing if it could give the same one back.

2. Set some Threshold to determine if the line is working efficiently

We ran into some problems here when we realized that we needed to incorporate some kind of function into our app. Because we can't incorporate percentages easily it was important for us to figure out various ways to write functions that would work in the same way.

3. Make the values get displayed back to the user in graphs and charts

The easiest part was making the values done in the previous two tests get sent back to the user allowing for them to see the suggested speeds the conveyors should be set too. The hard part was displaying it correctly eventually we got it to display the data the way we wanted

Updated Target Specifications:

Can Read and interpret data:

Read from the keyboard. Provide suggestions/ reflection. Accept different kinds of data.

Can do calculations:

Accept input and provide the result. Calculate the current speed and the ideal speed. Compare the two speeds.

Has a simple and easy to use UI

Simple windows and instructions for the windows. Use different colors to show positive and negative influences. Show with bold text and clear worlds.

Critical Systems

Calculations:

The Calculation system refers to all the calculation based aspects of the code needed for the app; this includes the percentage calculations, and the other value calculations that will be added to the next prototypes.

User Interface:

The user interface system refers to all the interactable or readable things our app has including buttons, text entering or visible data. The user interface is the most important part of our project as it should be simple and easy to use.

Input and Output:

The input and output system refers to all the data that will be stored or entered and displayed. The inputs will be the values that the user gives and the outputs will be the values after the value is run through various calculations. We will have multiple output values for only a few input values. It is also important that our input and output data can be stored as well as it makes it easier when the user is keeping track of the data.

Physical Component:

Some physical prototype that reads data from the conveyer in CPM must be replicable and work very efficiently.

Prototype II:

https://x.thunkable.com/copy/592e1810b87e90193a40222a93906f12

Prototype III Test Plan:

Test ID	Test Objective (Why)	Description of Prototype used and of Basic Test Method (What)	Description of Results to be Recorded and how these results will be used (How)	Estimated Test duration and planned start date (When)
1	Make the UI easy to use and visually pleasing. This will help to make our app easier to navigate and use practically. In order to achieve success in this portion we must ensure that no confusion is generated when using our app	Type: Focused and Physical To make the best UI possible it is important that we have some Input from people using the software for the first time. To test our prototype we are going to have someone navigate through the software while they are continuously speaking. Meanwhile, we are recording the struggles/confusions they may be having. Skills: Might take some time to figure out depending on what their preferences are.	We will take notes of the tester's responses to the UI. We will then use this data to change various things based on their struggles/suggestions.	Time: 30 min Date: Mar 15th 2023 Dependencies: After all other tasks have been completed this can be done this may need to be redone in the future as more interactions are added. Stopping Criteria: When we Think the UI is at an exceptional level.

2	Add some visuals to make entering the data easier. This will help the user when they are navigating the app and will allow for more flexibility.	Type: Focused and Physical Often, we find that simple visuals are very important so to test their effectiveness in our prototype we will use a set of flashcards on someone who has never seen the emoticons or visuals before. We will ask them if they understand what the visual is saying and if they understand very clearly then we will incorporate it into our prototype. If they don't understand it may be more confusing to include it. Skills: Should be easy to incorporate. materials:Thunkable, a person, and some flash cards	We will be taking notes based on the responses that our tester gives us. We can determine whether the images or symbols are too complex for them to understand and therefore not incorporate into our prototype.	Time: 20 min Date: Mar 15th 2023 Dependencies: None Stopping Criteria: When we finish implementing and testing the Criteria necessary for visuals.

3	Add a History section to see the previous values that were entered. Having a history section will help the managers to either repeat speeds calculations with ease or even help document line efficiencies with daily graphs. The most important part of this is to ensure that the previous data that was used is being saved to the history page.	Type: Focused and Analytical To incorporate some kind of history, function our app must be able to save previous data to some location on our screen. We will test this by setting a location for the history window and setting up a function that will add values to this section. This test will mostly be trial and error with different codes until we have multiple history values saving. Skills: Might take some time to figure out and might not be possible. materials:Thunkable	We will take notes of the trials we do. This information is not very important for our project but could be helpful for future tasks with similar functionalities.	Time: 1 hour Date: Mar 15th 2023 Dependencies: None Stopping Criteria:We will stop if we run out of time, cannot implement a function like this or we complete the history function.

4	Create a physical prototype that can read data and turn it into speed data.	Type: Physical, Analytical and Focused To run this test we will need to first develop the code that will be used to run the laser. Once this is working we can then physically attach the components to the required slots and then ensure that our code and the components run properly together .	We will ensure that the code and physical prototype have been set up properly to ensure that our physical prototype reads data and turns it into speed data. We will document the final codes and part locations as they are essential to making sure the code runs as intended.	Date:Mar 15th 2023 Dependencies: We will complete this last this can be done if we don't run out of time the app is our #1 priority.

References and Wrike:

https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=aMFdbngxqv2m6pfb6FRfTXzbfORiJzYj%7CIE2DSNZVHA2DELSTGIYA